



دانشکده علوم ریاضی

اصول پردازش تصویر

نیم سال اول ۱۳۹۹-۱۴۰۰

مدرس: دکتر مصطفی کمالی تبریزی

تمرین سری سوم - سوال اول

شماره دانشجویی: ۹۷۱۰۰۳۹۸

نام و نام خانوادگی: سیدعلیرضا خادم

زمان حدودی اجرا: کمتر از ۵ ثانیه

موارد لازم.

برای اجرا لازم است تا فایل Points.txt در مسیر EX3_Q1/ قرار داشته باشد. همچنین در پیاده سازی این سوال از کتابخانه های pandas، sklearn و matplotlib استفاده شده است که قبل از اجرا بایستی این کتابخانه ها روی سیستم شما نصب باشد.

روند کلی حل.

همانطور که در صورت سوال خواسته شده با استفاده از تابع matplotlib.pyplot.scatter نمایش نقاط در فضای دو بعدی را با نام res01.jpg در مسیر EX3_Q1/results ذخیره می کنیم. نمایش مجموعه نقاط موجود در فایل Points.txt تقریباً به صورت ۲ دایره هم مرکز است. با اجرای الگوریتم k-means روی این مجموعه نقاط، با توجه به نوع آرایش نقاط در فضای دکارتی و با توجه به اینکه معیار شباهت فاصله اقلیدسی نقاط در نظر گرفته شده است، همانطور که قابل پیش بینی بود و در شکل های 0.jpg تا 9.jpg نیز قابل مشاهده است مجموعه نقاط به صورتی کلاستر بندی می شوند که دو طرف قطر دایره بزرگ قرار می گیرند. حال اگر نقاط را به فضای قطبی ببریم، نمایش آنها به صورتی که در شکل 10.jpg در مسیر EX3_Q1/results/ قابل مشاهده است در می آید. نتیجه اجرای الگوریتم k-means روی نقاط در فضای قطبی به صورتی که در شکل res03.jpg مشاهده می کنید خواهد بود.

توضیح کد.

برنامه در مجموع حاوی ۲ فایل با فرمت py. می باشد که توضیحات هر فایل در پایین آمده است.

`data) number_of_points, load_data(points_file_data,`

این تابع مواردی که مشاهده می‌کنید به عنوان ورودی از کاربر می‌گیرد و بعد چاپ پیام polar/cartesian
 ”!!“ از کاربر می‌خواهد تا مود نمایش نقاط را مشخص کند. نقاط را از فایل ورودی می‌خواند و L در صورتی که کاربر polar
 را وارد کند مختصات نقاط در فضای قطبی و اگر cartesian را وارد کند مختصات نقاط در فضای دکارتی در data ذخیره
 میکند.

q1.py ○

در این فایل ابتدا فایل Points.txt از مسیر EX3_Q1/ باز می‌شود سپس تعداد سطرهای فایل که نشان دهنده‌ی تعداد
 نقاط هستند در متغیر num_of_points ذخیره می‌شود. در ادامه با استفاده از تابع load_data مختصات نقاط
 لود شده (با توجه به ورودی کاربر می‌تواند نقاط در مختصات قطبی یا دکارتی لود شده باشد) را در data ذخیره
 می‌کنیم. با استفاده از تابع pandas.DataFrame متغیر data_fram را محاسبه می‌کنیم و به عنوان ورودی تابع
 sklearn.cluster.KMeans.fit از آن استفاده می‌کنیم. در نهایت هم نتیجه خوشه‌بندی را با استفاده از توابع
 matplotlib.pyplot.scatter ، matplotlib.pyplot.show نمایش می‌دهیم.